HAIR: Hierarchical Visual-Semantic Relational Reasoning for Video Question Answering

Supplementary Material

Fei Liu[1,2] Jing Liu[1,2] Weining Wang[1] Hanqing Lu[1,2]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
liufei2017@ia.ac.cn {jliu, weining.wang, luhq}@nlpr.ia.ac.cn

A. Statistics of Datasets

In Table 7, we summarize the statistics of four VideoQA datasets used in our experiments. **TGIF-QA** dataset includes four tasks: *Action*, *Transition (Trans.)*, *FrameQA*, and *Count*. *Action* and *Trans.* are of multiple-choice task with five answer choices per question, and *FrameQA* and *Count* are of open-ended task. **MSVD-QA** and **MSRVTT-QA** datasets only contain open-ended questions with a pre-defined answer set of size 1,000. Questions are simply divided into five types (*i.e. what, who, how, when* and *where*) according to the initial word. **Youtube2Text-QA** dataset consists of the videos from MSVD video set and the question-answer pairs collected from Youtube2Text video description corpus. Both open-ended and multiple-choice tasks exist.

Table 7. Statistics of the VideoQA datasets used in our experiments. #MC: the number of answer choices for multiple-choice questions.

Dataset	#Video	#Question			Vocab. size	Ans. size	#MC
		Train	Val	Test			
TGIF-QA	71,741	125,473	13,941	25,751	8,000	1,746	5
MSVD-QA	1,970	30,933	6,415	13,157	4,000	1,000	NA
MSRVTT-QA	10,000	158,581	12,278	72,821	8,000	1,000	NA
Youtube2Text-QA	1,970	88,350	6,489	4,590	6,500	1,000	4

B. Visualization Results

We show more visualization results including some failure examples in Figure 8. It is seen from successful examples (shown on the left) that our model is able to focus on crucial objects and frames, and capture appropriate object-level and frame-level relationships, thus giving the correct answer. On the right we show some failure examples. The analysis is given as below.

In the first example, the answer-relevant object *arm* is hard to detect compared with the conspicuous object *hand* (or *finger*). Besides, when the man bounces arms, the hands are also bounced jointly. Therefore, the model tends to predict the answer related to *hand* (or *finger*). In the second example, since a regular explosion is usually followed by many scattered and asynchronous explosions, it is particularly difficult for models and even humans to precisely count the number of fireworks explosion. For the third example, although successfully retrieving the relevant objects and frames, the model mistakenly recognizes the key object *mower* as *car*. We find that *mower* is a rare answer in the dataset, and is similar to *car* in appearance. Therefore, the model may require more samples that have "mower" as the ground-truth answer to learn the fine-grained distinctions between mower and car. In the last example, the given video frames are rotated 90 degrees. This may result in that the action of "fall" is mistaken for the inverse action (*e.g.* "lift"), thus affecting the decision of the model. We believe these analysis of failure examples may inspire the future research on VideoQA.

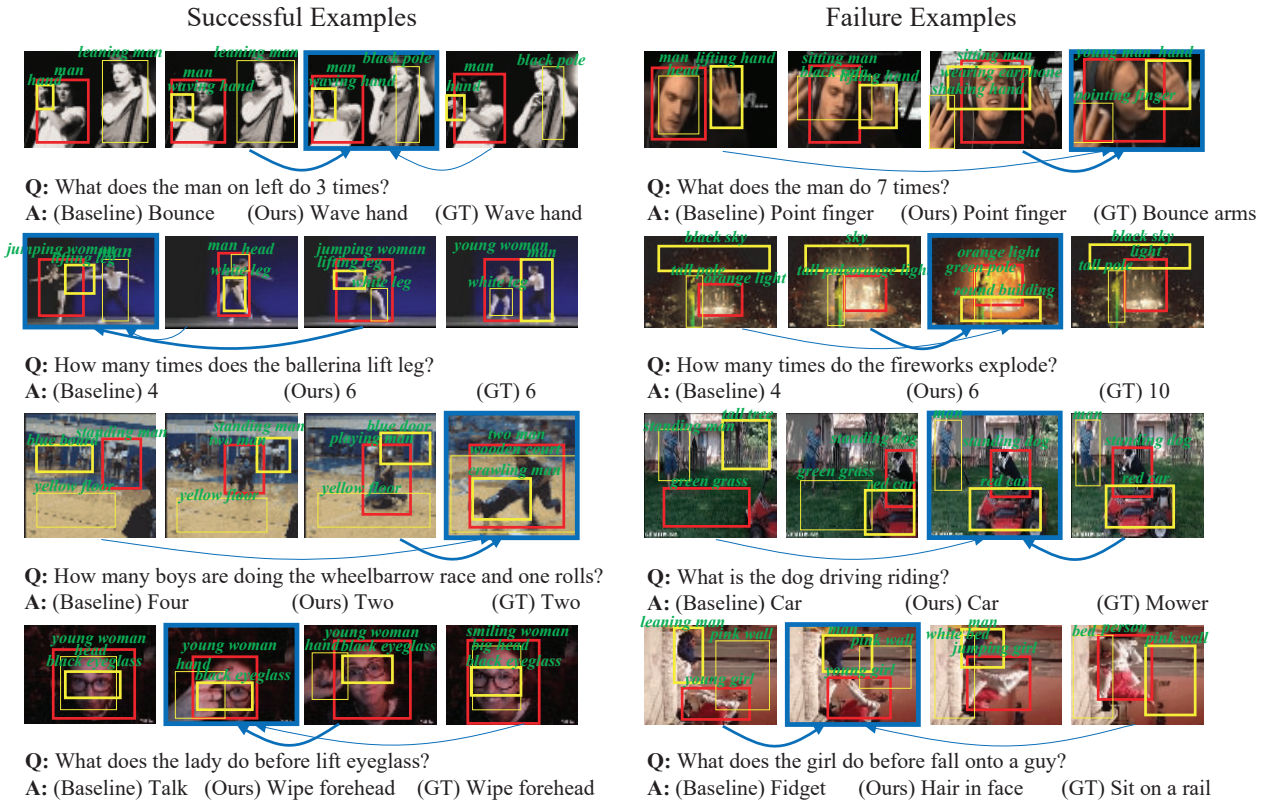

Figure 8. Visualization results generated by our HAIR. Successful examples are shown on the *left* and failure examples are on the *right*. GT: Ground-Truth.

C. Number of Detected Objects

We investigate the impact of different number of detected objects in Table 8. $N = 6$ attains the best performance on most tasks. Thus, by default, we use 6 detected objects per frame in experiments. $N = 2$ achieves the worst performance because some important objects may be missed. Using too many objects (*e.g.* $N = 10$) produces slight performance drop on most tasks. This may be due to that most samples in the dataset only rely on several salient objects to answer the question and feeding too many objects into network would confuse the model.

Table 8. Comparison of different number of detected objects on TGIF-QA.

#objects per frame	Action	Trans.	FrameQA	Count
$N = 2$	76.0	81.5	58.6	4.08
$N = 6$	**77.8**	**82.3**	60.2	**3.88**
$N = 10$	77.2	82.1	**60.9**	3.93

D. Multi-Scale Node Aggregation

For the VideoQA task, answering different questions usually needs temporal information of different durations. For example, answering the question *"How many times do the fireworks explode?"* may need information across the whole video, while answering the question *"How many boys are doing the wheelbarrow race and one rolls?"* may only need information from a few key frames. To this end, we develop a multi-scale node aggregation method to model the temporal information of different durations. The module consists of H heads, in which the core component is a 1D temporal average pooling layer with specific kernel size. We experiment with different H. As shown in Table 9, incorporating the multi-scale temporal information significantly improves the performance on the tasks that require strong temporal reasoning, such as

Action and *Trans.* tasks. It is not surprising that slight performance improvement takes place on the *FrameQA* task, where a single frame is sufficient to infer the answer.

Table 9. Comparison of different number of heads in the multi-scale node aggregation on TGIF-QA.

#heads (kernel sizes of temporal pooling)	Action	Trans.	FrameQA	Count
$H = 1$ (1)	75.3	80.6	59.5	4.08
$H = 2$ (1, 2)	76.8	81.7	60.0	3.97
$H = 4$ (1, 2, 3, 4)	**77.8**	**82.3**	**60.2**	**3.88**

E. Prediction Examples on MSRVTT-QA

In Figure 9, we show some prediction examples on the MSRVTT-QA dataset that contains longer video sequences. On these examples, our model gives the correct answer.

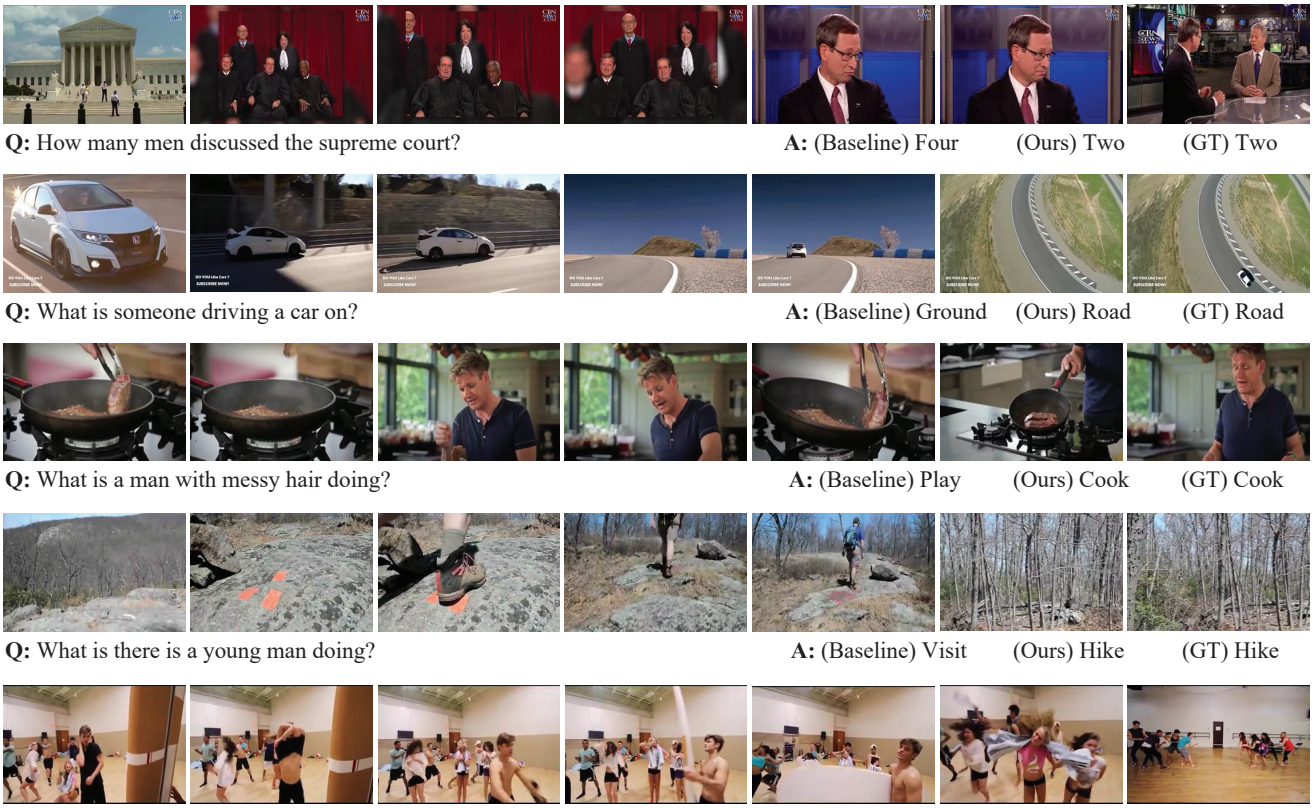

Figure 9. Prediction results on MSRVTT-QA. GT: Ground-Truth.